CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/68266/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Gao, Lin, Cao, Yan-Pei, Lai, Yu-Kun , Huang, Hao-Zhi, Kobbelt, Leif and Hu, Shi-Min 2015. Active exploration of large 3D model repositories. IEEE Transactions on Visualization and Computer Graphics , pp. 1390-1402. 10.1109/TVCG.2014.2369039

Publishers page: http://dx.doi.org/10.1109/TVCG.2014.2369039

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

Active Exploration of Large 3D Model Repositories

Lin Gao, Yan-Pei Cao, Yu-Kun Lai, Hao-Zhi Huang, Leif Kobbelt, Shi-Min Hu

Abstract—With broader availability of large-scale 3D model repositories, the need for efficient and effective exploration becomes more and more urgent. Existing model retrieval techniques do not scale well with the size of the database since often a large number of very similar objects are returned for a query, and the possibilities to refine the search are quite limited. We propose an interactive approach where the user feeds an active learning procedure by labeling either entire models or parts of them as "like" or "dislike" such that the system can automatically update an active set of recommended models. To provide an intuitive user interface, candidate models are presented based on their estimated relevance for the current query. From the methodological point of view, our main contribution is to exploit not only the similarity between a query and the database models but also the similarities among the database models themselves. We achieve this by an offline pre-processing stage, where global and local shape descriptors are computed for each model and a sparse distance metric is derived that can be evaluated efficiently even for very large databases. We demonstrate the effectiveness of our method by interactively exploring a repository containing over 100K models.

Index Terms—semi-supervised, active learning, data-driven, exploration

1 Introduction

With the rapid development of 3D acquisition and modeling techniques, geometric models have proliferated in recent years. Large model repositories exist such as Trimble/Google 3D warehouse [1], TurboSquid [2] etc. which contain hundreds of thousands or even millions of models. It is challenging to obtain an overview of these large repositories containing models of various categories and to find the best matching models for a given response. Previous efforts mainly took two directions: shape retrieval and data-driven exploration. Shape retrieval is essential to find the most similar models in a database, but does not support interactive exploration with the goal to get an overall idea of all relevant models in the repository. The effort the end user needs to invest in order to find a model of interest increases substantially when the repository scales up, as potentially a large number of very similar models exist. The data-driven approach, on the other hand, allows the user to explore the models by exploiting the relationship between models and thus gives a better overall feedback of relevant models. However, state-of-the-art methods focus on *small* datasets (typically tens to hundreds) all coming from the *same* category.

In this paper, we propose a novel approach that integrates these two directions to enable the interactive exploration of large multi category model repositories using an intuitive, data-driven approach. We argue that *exploration* is a more effective way of accessing and assessing huge amounts of information in the model repository than traditional model retrieval. By analyzing the relationship between database models in the offline stage, we make the online stage efficient for interactive exploration, even with a very large database. To the best of our knowledge, this is the first work that provides a data-driven exploration of a very large database. More specifically, the contributions of our paper are as follows:

- We propose a *scalable* approach to explore *large* model repositories dynamically, which involves manageable offline precomputation and *interactive* exploration in the online stage by maintaining a dynamic set of active candidates.
- We apply an *active learning* approach that allows users to *flexibly* explore models of interest by labeling entire models or parts of them as "like" or "dislike".
- We exploit the relationship between models in the repository and present them to the user in a relevance driven *parametric* space for *intuitive* exploration.

Fig. 1 gives an example of interactively exploring different chair models in our repository with over $100K$ models. The user starts by providing some initial query, in this case drawing a sketch (a) to roughly

- *L. Gao, Y.-P. Cao, H.-Z. Huang and S.-M. Hu are with the TNlist, Tsinghua University, Beijing 100084, China. E-mail: {gaolinorange,caoyanpei,huanghz08}@gmail.com, shimin@tsinghua.edu.cn.*
- *L. Gao is now with Beijing Key Laboratory of Mobile Computing and Pervasive Device, ICT, CAS. This work was performed while L. Gao was a PhD candidate at TNlist, Tsinghua University.*
- *Y.-K Lai is with School of Computer Science & Informatics, Cardiff University, UK. E-mail: Yukun.Lai@cs.cardiff.ac.uk.*
- *L. Kobbelt is with Computer Graphics Group, RWTH Aachen University, Germany. E-mail: kobbelt@cs.rwth-aachen.de.*

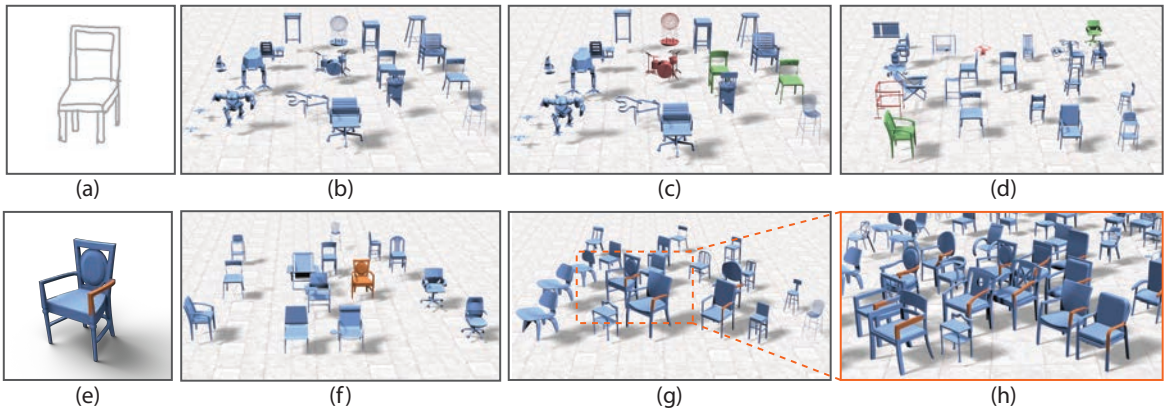


Fig. 1. Interactive exploration of chairs. (a) user input (sketch), (b) initial models retrieved, (c) with user preference highlighted (green for liked models and red for disliked ones), (d) result of global active learning, (e)(f) selected region of interest, (g) results of local active learning, (h) more models are revealed when zooming in.

express what models she is looking for. Alternative initial queries such as an example model can also be used. As no keywords are assumed in the large repository, sketch based retrieval might also return irrelevant models (b). Models are organized in a 2D parametric space where similar models tend to be placed closer to each other. Instead of showing all the retrieved models, *representative* models are shown with important models rendered larger. Users are then allowed to choose models they like (in green) or dislike (in red) (c), and the retrieved models are dynamically updated to reflect these preferences (d). The user is also allowed to choose certain regions of interest from a model (selected model in (f) with highlighted region in (e)). The exploration result is updated in (g) with models containing a similar chair handle emphasized. The local geometric information is leveraged to align the models such that the region correspondence can be derived easily. Our system also allows the user to zoom in within a certain region to discover more models (h), which were initially hidden due to the limited screen space. Details of the algorithm pipeline and the experimental setup are discussed in the following sections. The accompanying video shows interactive exploration using our system.

2 Related Work

To handle repositories with large numbers of models, shape retrieval has been extensively studied in recent years. Please refer to [3] for a comprehensive survey. For retrieval systems various global features have been proposed to compactly represent 3D models, such as shape distribution [4], spherical harmonic descriptors of the Gaussian Euclidean Distance Transform (GEDT) function [5], light field descriptors [6] etc. Such systems often use a combination of text, 2D sketches [7], [8] and 3D models as input to retrieve models with similar features.

Most shape descriptors are invariant to rigid transformations and thus are suitable for retrieving whole models. More refined shape descriptors are required to retrieve objects with partial similarity using boundary rasterization [9], compact local features and voting [10] and bag-of-features [11]. Bronstein et al. [12] propose an approach that uses intrinsic multiscale diffusion heat kernels [13] as local features to allow for retrieval of shapes with isometric deformation. For man-made objects, an approach is proposed based on a small set of predefined primitives and a probability model representing the spatial relationship [14]. Shape retrieval has also been demonstrated as a useful tool for modeling. Funkhouser et al. [9] propose a system for interactive shape modeling using geometric details from models in a database. To reduce the effort of user interaction, Xie et al. [15] interactively retrieve shape parts using sketches and assemble them for modeling. Xu et al. [16] on the other hand take a complete sketch describing a scene of objects as input and automatically construct the scene of 3D models, by exploiting the relationships between models.

While substantial effort has been made in shape retrieval, effectively finding relevant models from a large model repository is still challenging. The most liked models may not be returned as most relevant, merely based on user input and geometric signatures. Alternatively, the user may not have a clear idea at the beginning of the search process, and the current model search engine typically returns a list of models without organizing the content in a meaningful manner. It can thus be a frustrating process to browse through a long list of models to get the idea of relevant models and find the ones of interest. To address these issues, our approach provides flexible tools for users to express their preferences and not only retrieves relevant models, but also organizes them in a way that better captures the range of models in the repository and their relationships.

Semi-supervised learning for retrieval. To further improve the results of retrieval, user guidance has been used. Zhu et al. [17] propose a simple and efficient approach for active learning based on prob-

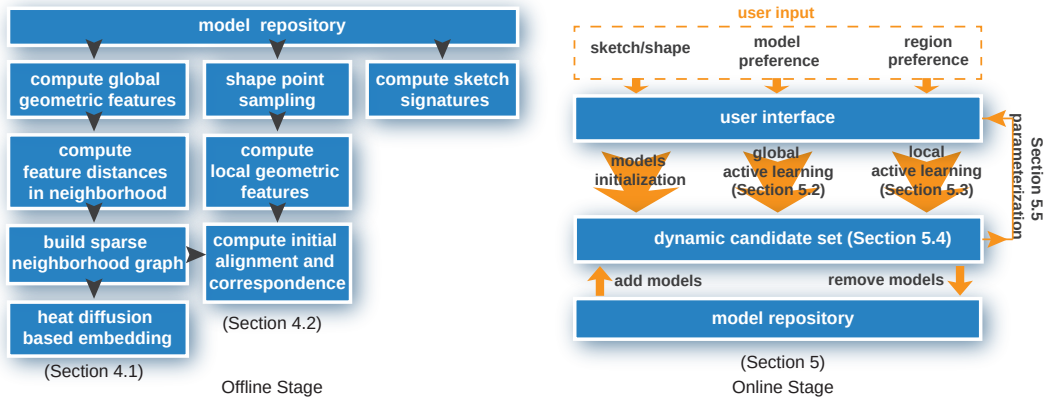


Fig. 2. System pipeline.

ability distributions on graphs. Semi-supervised techniques have been used in image retrieval [18], [19], [20] . In shape retrieval, relevance feedback [21], [22], [23], [24], [25], [26] has been used to bridge the gap between geometric features and semantics. Their approach is based on supervised feature extraction, i.e. using user input of liked or disliked examples to find an improved feature space for better discrimination.

Data driven shape analysis. The availability of large model repositories also helps shape analysis. Wang et al. [27] use active learning to segment shape sets in a semi-supervised manner. Consistent semantic labeling is obtained with a sparse set of user constraints. Huang et al. [28] use a semi-supervised approach to categorize shapes in a collection into fine-grain classes. To explore models of the same category, Ovjanikov et al. [29] propose a method to manipulate a template shape to explore similar shapes. Kim et al. [30] also use a template to learn the part-based variance of the model set. Kim et al. [31] use fuzzy correspondence to align models and select corresponding regions of interest on models in the dataset. These data-driven techniques provide more intuitive interaction and give a better idea of overall model distribution. However, they are generally restricted to a relatively small number of models (typically hundreds although [30] is able to cope with a few thousand models) due to the expensive co-analysis and more importantly only applied to models of the *same* category for co-analysis to be robust. Huang et al. [32] use functional map networks [33] to jointly analyze collections of similar shapes. Huang et al. [34] use the category tree to organize a heterogeneous collection of models for overview and exploration. While able to cope with models of different categories using a qualitative distance measure, the time complexity is $O(n^2)$ where n is the number of models, and processing large repositories would be prohibitively expensive. Kleiman et al. [35] propose an approach that organizes shapes in a dynamic 2D grid for model browsing.

To enable effective exploration of large model repositories, our approach is based on active learning, which unlike relevance feedback, takes into account both feature similarity to the user specified models and the potential effect of the user expressing their preferences. While effective, active learning typically requires $O(n^2)$ computation, which is not scalable to large model databases. With the aim of exploration of large number of models, we improve upon existing generic approaches by dynamically maintaining a small set of models for active learning, exploiting the relationship of geometric models and based on this, presenting the models in a parametric spatial embedding. The idea of using a parametric space is related to the work by Talton et al. [36] where they explore the parameter space to create new models for casual users. A small number of landmark models are used in their work while our approach deals with exploration of large number of models in the repository in a hierarchical manner.

3 System Overview

Our algorithm allows users to efficiently find models of interest in a very large model repository through an intuitive graphical interface. Starting with a 2D sketch or a reference 3D model, the system returns an initial set of candidate models which are arranged according to their relevance and mutual similarity. The user can incrementally refine the query by labeling some of the candidates either entirely, or only local parts of them, as "like" or "dislike". This information is fed into an active learning procedure and the set of recommended candidate models is updated. The pipeline of our system is illustrated in Fig. 2.

The main technical challenge is to establish a distance metric between the models in the database where the computational and memory complexity scales (nearly) linearly with the number of models (instead of quadratically) and which is sufficiently expressive to reliably model the mutual similarity between arbitrary shapes. We achieve this by first deriving a *sparse distance matrix*, which only contains non-zero entries for models that are sufficiently similar. With this reduction we are not losing any relevant global similarity information since in practice, the distance measures for largely different 3D models are

not very meaningful anyway. In order to propagate the sparse distance metric to all pairs of models, we compute a heat diffusion embedding (cf. Sec. 4.1).

To also enable the labeling of local parts of candidate models, we have to establish local correspondences between all the models in the database. Again, we reduce the computational complexity from quadratic to linear by propagating correspondence maps through a sparse similarity graph (cf. Sec 4.2). Notice that all the distance and correspondence computation is performed in an offline preprocessing step and does not slow down the interactive exploration.

In the online stage, when the user draws a 2D sketch (or provides a 3D reference model), sketch-based [8] (or model-based [6]) shape retrieval techniques return a number of models similar to the input. Active learning is used to iteratively refine the obtained candidate models. To deal with a large number of models, a dynamic subset of relevant models is maintained. Models are organized in a 2D parametric space based on their mutual similarity so that the overall distribution of models can be perceived at a glance, which helps the user to better understand the distribution of relevant models and enter their preference (Sec. 5).

4 Scalable Model Similarity Metrics

For each model in the repository global and local metrics are precomputed, along with the sparse relationship between similar models.

4.1 Global Similarity Metrics

To make the global query operations effective, we use a global similarity matrix to represent the relationship between pairs of models. But to also guarantee efficiency, we only explicitly calculate similarities for models that are sufficiently similar, leading to a highly sparse matrix which is precomputed in the offline stage. In the online stage, we also propose cross voting for effective retrieval.

In this work, the light field descriptors (LFD) [6] are used as global features as they are known to be well suited for model retrieval [37]. Recent descriptors such as Heat Kernel Signatures are insensitive to isometric deformation, however such descriptors generally assume manifold surfaces, whilst many models available are of poor connectivity. To calculate light field descriptors, the models are scaled to fit in a regular dodecahedron and cameras are put in each vertex to get 2D images. Each image is converted into a 47-dimension signature, including Zernike moments and Fourier descriptors. The 20 images from a particular orientation of the dodecahedron form a complete set of descriptors to represent the model for this orientation. The light field distance between a pair of models is defined as the L_1 vector distance between the signatures after an optimal rotation is applied, making it robust to rotations [6].

Finding close models in the light field distance is expensive as thousands of combinations are considered for each pair of models. We propose a simple and efficient heuristic to suggest similar models. Based on the assumption that similar 3D models have well matched 2D views, we put signatures of every view of every model in a kd-tree and use it to efficiently suggest potential models for calculation of the light field distance. For each image of one model M_i, we find the nearest k images ($k = 6$ in our experiments) using the L_1 metric in the kd-tree, and take their corresponding models M_j as candidates. The light field distances between M_i and each M_j are calculated. Notice that the number of models M_j that need to be compared to M_i is typically much smaller than the theoretical maximum of $20 \times 6 = 120$ since several models appear multiple times in the k-nearest signatures lists. For each M_i we eventually keep the k closest models based on the light field distances. For the model pair M_i and M_j, we add both entries (i, j) and (j, i) in the sparse light field distance matrix S_d, leading to a symmetric matrix. The non-zero entries in S_d form a sparse graph G_d with edges connecting similar models. We further define the similarity matrix as follows:

$$W(i,j) = \exp\left\{-\frac{S_d(i,j)^2}{\sigma_1^2}\right\}, \tag{1}$$

where σ_1 is a scale parameter ($\sigma_1 = 4000$ is used throughout the paper). This gives a normalized similarity with 1 representing identical LFD and close to 0 if models are sufficiently different.

To quickly estimate the distance between an arbitrary pair of models, we follow the diffusion framework [38] which gives an optimal low-dimensional heat diffusion embedding as follows. Given the similarity matrix W representing the (unnormalized) transit probability between models, we first define $D = diag(\sum_j W_{ij})$ and the normalized matrix Z is defined as $D^{-1}W$. As discussed in [38], Z is similar to the real symmetric matrix $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, thus the eigenvalues of matrix Z are real values. With the normalization, the eigenvalues are distributed as $1 = \lambda_0 \geq \lambda_1 \geq .... > 0$. The eigenvector υ_0 corresponding to the first eigenvalue λ_0 is a vector containing identical entries. Discarding υ_0 corresponds to shifting the center of gravity of each model to the origin such that the diffusion coordinates ϕ_i for model M_i are obtained by taking the eigenvectors corresponding to the l largest eigenvalues:

$$\phi_i = \left(\lambda_1^t \upsilon_1(i), \lambda_2^t \upsilon_2(i), ..., \lambda_l^t \upsilon_l(i)\right), \tag{2}$$

where $\upsilon_j(i)$ is the i^{th} component of υ_j, t is the timescale of diffusion analysis, and l determines the dimension of the embedding space. $t = 10$ and $l = 50$ are used in all our experiments. The diffusion distance

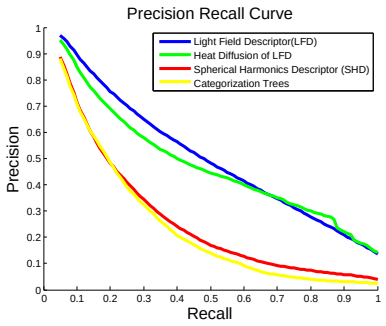


Fig. 3. Precision-recall curves of different shape features on the Princeton Shape Benchmark.

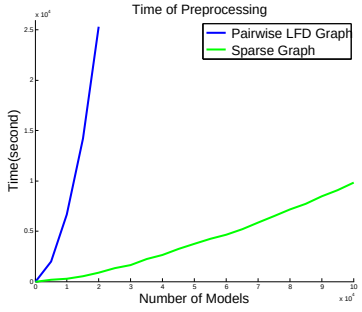


Fig. 4. Preprocessing times using original LFD (blue) and heat diffusion embedded LFD (green).

between two models M_i and M_j is then defined as the Euclidean distance in the embedding space $\hat{d}(i,j) = \|\phi_i - \phi_j\|_2$. With the embedding space precomputed, the diffusion distance is efficient to calculate. The nearest neighbors in the diffusion distance can be efficiently retrieved using kd-tree acceleration.

We performed quantitative comparisons of different shape descriptors using the retrieval tasks on the Princeton Shape Benchmark [37] with ground truth labels (Fig. 3). The precision-recall (PR) curves of the original light field descriptors (LFD), our heat diffusion accelerated embedded LFD, spherical harmonic descriptors (SHD) and the degree of separation (the number of edge hops between models) in the categorization tree (CT) [34] are shown. CT is more suitable for the exploration and quality measurement of heterogenous models and does not work particularly well for such data. Although the original LFD feature is effective for retrieval, distances need to be calculated between every pair of models, which takes quadratic preprocessing time and thus does not scale well to large model repositories. Moreover, it is demanding to store pairwise distances. With $O(n^2)$ memory cost, ordinary computers are not able to process large repositories. Our heat diffusion embedded LFD has similar performance as the original LFD, but can be calculated much more efficiently (nearly linear preprocessing time), making it suitable for large repositories, as shown in Fig. 4. Both descriptors are significantly better than SHD.

In the online stage, the user starts by giving some initial input to express their desired target. This can be a similar model or a rough sketch [39]. As will be shown later, even if the example model or sketch is quite far from the models of interest, our active learning approach is able to find the models with a small effort of user interaction. K (typically 200 depending on the number of models of interest) good matches are retrieved for exploration. As initial retrieval is not very robust, $2K$ models closest in sketch (or geometric feature) are first returned with a similarity score $h(i)$ representing how well the model matches the input, which may not be an accurate representation of similarity. Under the assumption that geometrically similar models have similar similarity scores and the majority of the models returned are correct, we propose a cross voting algorithm that produces the refined score $S(i)$ as the average of scores of similar models $h(j)$, weighted by similarity derived from the diffusion distance $\hat{d}(i,j)$,

$$S(i) = \sum_j h(j) \exp\left\{-\frac{\hat{d}(i,j)^2}{\sigma_2^2}\right\}, \qquad (3)$$

with $\sigma_2 = 0.001$ in our experiments. In principle all the $2K$ models are considered; however, due to the local support nature of the weights, only those models in the local neighborhood of model i in the feature space are effective. We keep K models with the largest refined score for subsequent consideration. An example is given in Fig. 5 where the top row shows the initial matches based on the drawn sketch (car), and the second row based on the refined score, both in descending order of the similarity scores. The effectiveness of cross voting is also quantitatively evaluated by measuring the precision (the percentage of correct models) in the K initially retrieved models from our large repository containing over $100K$ models, either directly or with cross voting. The recall is not feasible to obtain as assigning ground truth labels for all the models is impractical. As shown in Table 1, the precision improves significantly with cross voting.

Example	Directly (%)	With Cross Voting (%)
Chairs (Fig. 1)	53.5	70.5
Humans (Fig. 11)	55.0	64.0
Quadrupeds (Fig. 11)	13.0	29.5
Cars (Fig. 11)	85.0	91.0
Fighters (Figs. 9 and 12)	52.0	64.5

TABLE 1
Comparison of the precision of the initial retrieval results either directly or after using cross voting.

4.2 Local Similarity Metrics

Global properties provide a useful tool to measure the similarity of the whole models. In addition, local similarity helps to fine-tune the active learning based on local features. As before, we directly compute fuzzy correspondence only for similar models (Sec. 4.2.1) and the information is then propagated to an arbitrary pair of models. This ensures efficiency but also helps to improve robustness. The local correspondence is

Fig. 5. The obtained models without (top row) and with (bottom row) cross voting.

propagated using an augmented local graph with increased connectivity to ensure robust propagation (Sec. 4.2.2). This is then used to propagate user selected local regions to similar models, which are further refined using a Markov Random Field (MRF) approach (Sec. 4.2.3).

4.2.1 Alignment and Fuzzy Correspondence

We compute fuzzy correspondence between models based on the initial co-alignment derived from the light field calculation. We first apply furthest point sampling on each model using Euclidean distances to obtain m points ($m = 256$ in our experiments), and scale the model to fit within a unit bounding sphere. Euclidean distance is used for both efficiency and robustness to non-manifold meshes. Rigid transforms that give small distances of light field signatures (following the calculation of light field descriptors) are used as candidates, and the alignment is obtained by finding the transform with the minimal overall distance between sample points.

Given two models, namely M_{src} and M_{tgt}, for every sample point p_i on M_{src}, we can find a nearest point $\bar{p}_{t_i}$ on M_{tgt} according to the initial alignment from the light field matching. We estimate a fuzzy distribution of correspondence for each vertex. The basis vector f_i for source vertex i is a Kronecker delta function $f_i(p_k) = \delta_{ik}$, which is 1 for $k = i$ and 0 otherwise. The corresponding distribution on M_{tgt} is estimated based on geometric and feature closeness. For any sample point $\bar{p}_k$ on M_{tgt} within a sphere centered at $\bar{p}_{t_i}$ (with a radius of 0.15 in our experiments) the unnormalized correspondence response is calculated as:

$$\bar{f}_i(\bar{p}_k) = G(\bar{p}_k; \bar{p}_{t_i}) S(\bar{p}_k; p_i), \tag{4}$$

where $G(\bar{p}_k; \bar{p}_{t_i})$ is based on geometric closeness to the estimated target, and $S(\bar{p}_k; p_i)$ is based on feature closeness to the source point. We define

$$G(\bar{p}_k; \bar{p}_{t_i}) = \exp\left\{-\frac{\|\bar{\mathbf{p}}_k - \bar{\mathbf{p}}_{t_i}\|^2}{\sigma_3^2}\right\}, \tag{5}$$

$$S(\bar{p}_k; p_i) = \exp\left\{-\frac{\|\bar{\mathbf{s}}_k - \mathbf{s}_i\|^2}{\sigma_4^2}\right\}, \tag{6}$$

where $\|\cdot\|$ is the L_2 norm, $\bar{\mathbf{p}}_i$ is the geometric coordinates of vertex i on M_{src} and M_{tgt} respectively, and $\mathbf{s}_i$ and $\bar{\mathbf{s}}_i$ are the geometric feature vectors at vertex i on M_{src} and M_{tgt}. We use unique shape context [40] due to its distinctiveness, and the distribution $\bar{f}_i$ is then normalized to sum to one. The unique shape context captures sufficient local geometric information despite a sparse sampling. $\sigma_3 = 0.4$ and $\sigma_4 = 7$ are used in our experiments.

Given each basis vector f_i on M_{src}, the mapped distribution $\bar{f}_i$ on M_{tgt} can be calculated using Eqn. 4. The fuzzy correspondence map $T_{src\to tgt}$ maps an arbitrary distribution on M_{src} to a distribution on M_{tgt}. Assuming this mapping is linear, it satisfies

$$T_{src\to tgt}(f_1, f_2, \cdots, f_m) = (\bar{f}_1, \bar{f}_2, \cdots, \bar{f}_m), \tag{7}$$

where $T_{src\to tgt}$ is the matrix representing the mapping, f_i and $\bar{f}_i$ are column vectors, $(f_1, f_2, \cdots, f_m)$ and $(\bar{f}_1, \bar{f}_2, \cdots, \bar{f}_m)$ form two matrices. Since $(f_1, f_2, \cdots, f_m)$ is an identity matrix, $T_{src\to tgt} = (\bar{f}_1, \bar{f}_2, \cdots, \bar{f}_m)$. Given a subset of sample points γ on M_{src}, a distribution function f can be defined as a vector with $f(i) = 1$ if $p_i \in \gamma$ and $f(i) = 0$ otherwise. The corresponding distribution vector $\bar{f}$ on M_{tgt} can be obtained as $\bar{f} = T_{src\to tgt} f$. If an edge exists in the sparse graph G_d between a pair of models M_i and M_j we precompute $T_{i\to j}$ and $T_{j\to i}$ in the offline stage.

4.2.2 Correspondence Propagation using N-order Graph

During the interactive exploration, a set of K models are maintained as active candidates. A subgraph G of the global sparse graph G_d is then extracted from the database, keeping those vertices corresponding to models in the active candidate set and their adjacent edges. As establishing correspondences between significantly dissimilar models would be unreliable, we rather propagate fuzzy correspondences calculated for neighboring models (as in the previous subsection). However, when the path connecting two models is too long, the propagation quality may also drop. We thus propose to augment the subgraph G to obtain a graph $\tilde{G}$ with stronger connectivity to robustly propagate fuzzy correspondence between models. Let us denote by V the nodes of G. For a given source node s, the order of a vertex v is defined as the length of the shortest path from v to s. An N-order graph $\tilde{G}_s$ w.r.t. the source s satisfies that for every node $v \in V$, there exists a path $\mathcal{P}$ from v to s with its length (the number of edges) $|\mathcal{P}| \leq N$.

We use a greedy approach to incrementally add new edges to G. For this purpose, the order of each node can be obtained by breadth-first traversal from s. We maintain two sets of nodes V_1 with all the nodes of order less than N and V_2 with all the nodes of order more than N. We find $v_i \in V_1$ and $v_j \in V_2$ such that the light field distance between v_i and v_j is minimum. Light field distances are only calculated on a candidate set of pairs which are efficiently obtained using diffusion distances. After adding the edge (v_i, v_j), the node orders are updated. We compute the fuzzy correspondence map between v_i and v_j directly using the method in Sec. 4.2.1. Since adding the edge (v_i, v_j)

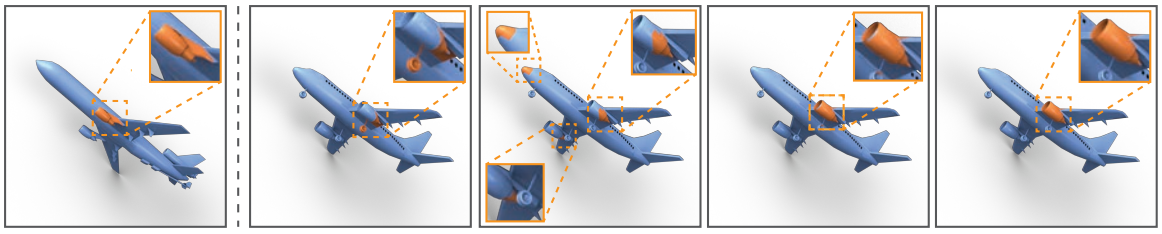


Fig. 6. An example demonstrating local correspondence. (a) an engine selected by the user from one model. (b-d) Corresponding regions in another model using thresholding of fuzzy correspondence response, with only geometric closeness (b), feature closeness (c) and both (d). (e) the final region after the MRF-based optimization.

makes v_j of order N or less, v_j (possibly also some of v_j's neighbors) will be removed from V_2. With at most $|V_2|$ iterations, the set of V_2 will become empty and the obtained graph is N-order.

Given a shortest path $p : s, i_1, i_2, \ldots, i_u$ of length u, we can obtain the propagated correspondence map

$$T_{s\to i_u} = T_{i_{u-1}\to i_u} \ldots T_{i_1\to i_2} T_{s\to i_1}. \quad (8)$$

As demonstrated later, setting N too large or too small both lead to less robust results. $N = 2$ generally works well and is used in our experiments.

4.2.3 Part Selection Propagation

To allow local active learning, user selection is propagated to other models as follows. When the user selects a part on a model M_{src}, we take M_{src} as the source node to build an N-order graph and obtain the propagated fuzzy correspondence map $T_{src\to tgt}$ to any model M_{tgt} in the current set. The selected region is represented as the source distribution f_{src} with 1 representing selected sample points and 0 unselected. The target distribution is simply obtained as $f_{tgt} = T_{src\to tgt} f_{src}$. f_{tgt} is a fuzzy selection and to obtain a discrete label l_p for each sample point p on M_{tgt} with $l_p = 1$ representing selected points and 0 otherwise, we use a Markov Random Field (MRF) that minimizes the following energy, taking into account both the local probability and the spatial closeness:

$$E(L) = \sum_p D_p(l_p) + \sum_{\{p,q\}} V_{pq}(l_p, l_q), \quad (9)$$

where $L = \{l_p | p \in P\}$ represents an assignment of labels and P is the set of sample points. The first term sums over all the sample points $p \in P$ with the penalty being the uncertainty of assigning label l_p to the sample point p, which is defined as $D_p(l_p = 0) = \min(f_{tgt}(p), 1)$, and $D_p(l_p = 1) = \max(1 - f_{tgt}(p), 0)$. The second term is a regularization term that conceptually sums over all pairs of sample points $\{p, q\}$ defined as $V_{pq}(l_p, l_q) = \exp\left\{\frac{-d(p,q)^2}{\sigma_5^2}\right\}(l_p - l_q)^2$. Due to the Gaussian local support nature of the weight function, only neighboring samples need to be considered in practice. $d(p, q)$ is the Euclidean distance between sample points p and q, and $\sigma_5 = 0.15$ is used in our experiments. The MRF-based optimization is efficiently solved using graph cut [41].

An example in Fig. 6 demonstrates local correspondence. The user selects an engine region on a plane model (a), and the corresponding regions on another plane model obtained by thresholding fuzzy correspondence responses are shown in (b-d). Using geometric closeness in Eqn. 4 alone, the obtained region in (b) does not align with the actual engine. Using features alone, the obtained region in (c) spreads over the plane including part of the engine on the other side. Using both geometric and feature closeness (d), the engine is correctly mapped. The result is further improved by using the MRF-based optimization (e). In this case, a small wheel part is successfully removed from the corresponding region.

5 Active Learning based Exploration

To effectively explore models in a large database, we use an active learning approach to iteratively refine the results based on user input. Unlike traditional active learning, our approach allows users to choose *models* or *parts* that they like or dislike, and retrieves and presents the updated set of suitable models at interactive rate. We first build an augmented local graph with sufficient connectivity to ensure robust active learning and visualization (Sec. 5.1). The user is allowed to specify their preference either based on whole models or selected parts. Global active learning is used to improve retrieved models based on whole models whereas local active learning is used for using exploiting local preference information. Details of global and local active learning are then given in Secs. 5.2 and 5.3 respectively. At any time we maintain a dynamic set τ of candidate models, starting with those returned from the initial search, which permits efficient exploration of the model database (Sec. 5.4). The user interface organizes the retrieved models in a 2D parametric domain, with the most relevant models emphasized. Thus the user can easily capture the overview of the relevant models, making user preference specification more efficient (Sec. 5.5).

5.1 Augmented Local Graph

Given the dynamic set of models τ and the existing edges from the global database (the sparse graph G_d),

as shown later in the section, we need a sufficiently well connected graph to allow efficient active learning and embedding for visualization. Let us denote G_τ as the subgraph of G_d containing nodes in τ and edges involving nodes in τ. We first augment the local graph G_τ with additional edges such that every model has at least $\bar{k}$ neighbors ($\bar{k} = 4$ is used in experiments). This is achieved by checking each model in turn and adding edges to the closest models according to the diffusion distance (due to its efficiency). After this, the obtained graph may still have multiple disjoint components. We use Prim's algorithm to construct a minimal spanning tree which connects the components. Edges are added incrementally between that pair of models belonging to different connected components with the minimal diffusion distance, until the graph is connected. We denote the augmented graph $\bar{G}_\tau = (\bar{V}_\tau, \bar{E}_\tau)$, where $\bar{V}_\tau$ is the node set and $\bar{E}_\tau$ is the edge set. In practice only a small number of auxiliary edges are added which does not have significant effect on the performance.

5.2 Global Active Learning

Let us denote y_i as the preference for the i^{th} model in the current model set τ. $y_i = 1$ (or 0) means the user likes (or dislikes) the model in this exploration. We relax y_i to be a real variable [17]. And denote $\mathbf{y}$ to be the vector of y_i's. Similar models usually have similar preference values. Global active learning finds a labeling by finding $\mathbf{y}$ that minimizes the following global energy over the augmented graph $\bar{G}_\tau$:

$$E_{\bar{G}_\tau}(\mathbf{y}) = \frac{1}{2} \sum_{i,j:(i,j)\in\bar{E}_\tau} w_{ij}(y_i - y_j)^2, \qquad (10)$$

where w_{ij} is the similarity between models M_i and M_j, as defined in Eqn. 1. The solution of Eqn. 10 is a harmonic field, which can be efficiently obtained by solving a linear system. When the user selects the liked and disliked models, the operation defines the Dirichlet boundary condition of the harmonic field.

5.3 Local Active Learning

To give the user more flexibility, we further propose local active learning which allows the user to select some part of a model and indicate whether this part is liked or not. Assuming model M_i is selected, for any model M_j we obtain the label on the model M_j using the maps T described in Sec. 4.2 which effectively selects a set of sample points on M_j. In order to measure the similarity between the selected part on M_i and corresponding part on M_j, we take precomputed shape context signatures [40] for *selected* sample points on models M_i and M_j, which are denoted as $\mathbf{C}_i$ and $\mathbf{C}_j$. Both matrices are of size $m \times q$, where m is the number of sample points and q is the dimension of the feature. The matrix entries are zeros for those rows related to unselected sample points. The feature distances between the fuzzy correspondence regions are calculated as

$$\bar{d}(i,j) = \|T_{i\to j}\mathbf{C}_i - \mathbf{C}_j\|_F, \qquad (11)$$

where $\|\cdot\|_F$ is the Frobenius norm of the matrix. From the function map point of view [42], the signatures are the functions defined on the sampling points and $T_{i\to j}$ is the function map. The bases are the indicator functions on the sampling points. If two models do not have a corresponding part, the shape context signatures will be very different. So the $\bar{d}(i,j)$ will be large. We add the following local energy terms to the function to be minimized (Eqn. 10):

$$E_L(y) = \frac{1}{2} \sum_j \exp\left\{-\frac{\bar{d}(i,j)^2}{\sigma_6^2}\right\} (y_j - v_j)^2, \qquad (12)$$

where v_j is 1 if the user likes the part (that penalizes small values) and 0 otherwise. $\sigma_6 = 6$ is used in experiments. Global preference is usually used along with local preference, and we thus minimize $E_{\bar{G}_\tau} + E_L$, which leads to a linear system that can be efficiently solved. An example is given in Fig. 7 showing the preferred models after selecting one chair globally (top row) or locally in the back of the chair, as highlighted (bottom row). Local active learning is more effective in expressing the preference for local regions, as the results show more chairs that have the similar back.

Fig. 7. Global (top row) vs. local (bottom row) active learning: models in descending order of preference y_i.

5.4 Dynamic Set

Since the model repository can be very large, direct application of active learning to the whole data set is prohibitively expensive. Our solution is to instead maintain a small dynamic set τ containing a subset of K models. In each interaction step, we propagate the user interests to the whole repository by allowing new relevant models to be added and irrelevant models to be removed dynamically.

After each interaction, we propagate the preference field y_i to the neighboring models in the *global* repository. For each model in τ, we find $\tilde{k}$ nearest models in the diffusion distance, accelerated using a kd-tree. We take $\tilde{k} = 10$ in this work. Assuming model M_j is one of the neighbors of model $M_i \in \tau$ currently being considered, the propagated preference value is defined as $p_{i\to j} = y_i \exp\left\{-\frac{\tilde{d}(i,j)^2}{\sigma_7^2}\right\}$,where $\tilde{d}(i,j)$ is the diffusion distance between M_i and M_j. $\sigma_7 = 0.01$

in our experiments. The total propagated value for M_j is the sum of propagation received from all the neighbors in τ, i.e. $p_j = \sum_{i:M_i \in \tau} p_{i \to j}$.

The dynamic set τ is first updated by adding T new models with the largest propagated values. We set $T = \frac{K}{4}$ in our experiments, where $K = |\tau|$ is the size of the dynamic set. Given the same user preference as boundary conditions, we obtain the updated preference values by minimizing the active learning energy $E_{\bar{G}_\tau}$ or $E_{\bar{G}_\tau} + E_L$ like described in the previous section. The models with lowest values except for the user labeled ones will then be removed. This maintains a consistent number of active models and ensures interactive performance. While previous co-analysis based exploration methods may be able to handle a repository containing K models, they are not directly applicable because the dynamic set is updated after each user interaction, and such methods typically require minutes for pre-processing, given a new set of models. Also, most methods are designed for models of the same category which is not generally satisfied for τ.

5.5 User Interface Design

In order to present the active set of candidate models to the user in a way that intuitively conveys the current state of the exploration session, we develop a user interface that is designed to provide as much information as possible without generating excessive visual clutter. Our interface concept is based on presenting the 3D models over a 2D plane while their relative position and size encode their similarity and relevance respectively.

Position: When the user is confronted with an unsorted set of candidates, it is difficult to obtain a structured assessment of what types or classes of models are available. Hence, we determine the relative position of the models such that similar models are grouped closer together. This can be achieved by applying the Isomap [43] algorithm to compute a 2D embedding of the models that locally preserves their relative distances from the light field descriptor space.

Size: To express the different degrees of relevance for the refinement of the exploration, we scale the models according to their preference but also by their potential to have significant impact on the candidate set update when they are labeled as "like" or "dislike". For this we use a measure for the risk of misclassification (= uncertainty), and the risk of each model is defined as the modified total risk when the model is assigned a label. The labeling of low-risk models by the user has more impact on the exploration than the labeling of high-risk models since for high-risk models the labeling is uncertain while low-risk models more effectively reduce the ambiguity among the candidates. Hence we display models with high preference and low risk larger. More specifically,

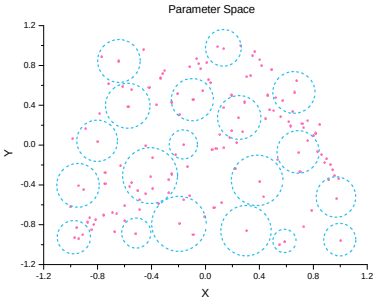


Fig. 8. The parameter space for visualization. Each red dot corresponds to a model and blue circles indicate representative models and their sizes.

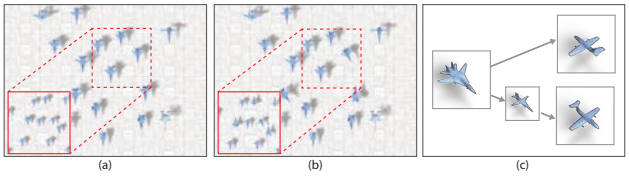


Fig. 9. Exploration of fighters. (a) alignment with $N = 2$, (b) alignment with $N = 5$, (c) direct model alignment vs. via an intermediate model. The solid red rectangle corresponds to the dashed red rectangle when zoomed in with more models revealed.

given a label assignment L, the risk is defined as

$$R(y) = \sum_i \sum_{v_i=0,1} [sgn(y_i) \neq v_i] p(y_i|L), \qquad (13)$$

where $sgn(x)$ is 1 if and only if $x > 0.5$, $[\cdot]$ is 1 if the condition is true and 0 otherwise, $p(y_i|L)$ is the probability of y_i given the label assignment L. The expected modified risk is

$$R_i = (1 - y_i) R(y|y_i = 0) + y_i R(y|y_i = 1), \qquad (14)$$

which considers the expected impact on the risk with the additional preference specified for the model i [17]. R_i is normalized by linear scaling to $[0, 1]$, denoted as $\hat{R}_i$. The size of the model M_i in the parametric domain is determined by a radius r_i, defined as $r_i = r_0 \left((1 - \lambda) y_i + \lambda (1 - \hat{R}_i) \right)$, where λ is a constant balancing the weights of both terms ($\lambda = 0.3$ in experiments), r_0 is a global scaling factor set to 0.3, with the 2D parameter space normalized to $[-1, 1]^2$.

Clustering: Finally, in particular in large model repositories, we often find a large number of very similar models which can lead to visual clutter, making it difficult for the user to keep a good overview. This is why we apply a clustering scheme that replaces groups of similar models by one representative each. Only if the user zooms in on one of these representatives, the group members are displayed to enable a refined exploration.

We maximize the following function to decide

Example	t_g (s)	t_l (s)	K
Chairs (Fig. 1)	0.745	2.640	200
Humans (Fig. 11)	0.779	1.789*	200
Quadrupeds (Fig. 11)	0.732	1.655*	200
Cars (Fig. 11)	0.429	0.998*	100
Fighters (Figs. 9 and 12)	0.756	2.940	200

TABLE 2
Statistics of different exploration. t_g, t_l: running times with global as well as global and local active learning. K: dynamic set size. *for alignment only.

which models are to be selected for visualization:

$$E_V(x) = \sum_i \left((1-\lambda)y_i + \lambda(1-\hat{R}_i)\right) x_i,$$
$$s.t. \quad \forall i, \sum_{\|\mathbf{P}_i - \mathbf{P}_j\| < r_i + r_j} x_j = 1, \qquad (15)$$

x_i is a 0-1 variable indicating whether the model is selected as a representative for visualization. $\mathbf{P}_j$ is the 2D position of the model M_j in the parameter space. The constraint $\|\mathbf{P}_i - \mathbf{P}_j\| < r_i + r_j$ ensures that there is no overlap between two models in the parameter space (and 3D space). This is a 0-1 programming problem which is efficiently solved using a branch-and-bound algorithm [44]. It takes less than 0.12 second for all the examples in the paper. The parameterization and representative model selection are illustrated in Fig. 8 (corresponding to the example in Fig. 1(g)) where each red dot represents a model in τ and blue circles indicate the size of the bounding sphere for each representative model. Our method effectively finds representative models without overlapping in space. When the user zooms in, a portion of the 2D parameter domain is mapped to cover the whole view space, by linearly scaling the coordinates. The distances between models become larger. Thus more models can be revealed without overlapping.

6 RESULTS

Our experiments were carried out on a computer with a 2.4GHz Intel E5620 CPU and 16GB memory. Our model repository contains $103,738$ models from the following sources: $18,312$ models from Tsinghua 3D model repositories [45], $10,911$ models from the light field retrieval repository [6], $1,814$ models from the Princeton shape benchmark [37], $1,200$ models from SHREC'12 [46], 380 models from the Princeton shape segmentation benchmark [47]. The remaining models are from Google 3D warehouse [1]. Duplicate models (automatically detected as models with identical shape descriptors) are removed.

Running times and comparisons. The one-off preprocessing time for the repository after calculating geometric features is $9,524$ seconds (about 2.65 hours). For a repository with n models, finding the neighboring models takes $O(n \log n)$ time and computing the first few eigenvectors of the sparse matrix was performed using the Lanczos algorithm, which takes linear time in the nonzero elements ($O(n)$). The preprocessing thus takes $O(n \log n)$ and can be easily parallelized. This shows that our method scales well with large model repositories. For examples in the paper, the average time of the initial sketch-based retrieval is 1.72s. The total duration of each interactive session using global or local active learning are reported in Table 2. The average time of our global active learning is 0.75s in these examples, and the average time of Leng's relevance feedback method [24] is 1.62s. Note however their method is only for shape retrieval rather than exploration, and does not consider relationship between models.

Parameter settings. Our method has a few parameters; the fixed values reported in the paper were found empirically and worked well for all the examples presented. The only adjustable parameter is the number of models K in the dynamic set (200 by default), which is specified by the user depending on how much model variation the user prefers to exhibit at the same time. As shown in Fig. 10, with increasing K, more variation of models from the repository are presented, while the number of models displayed is more or less fixed, because this is restricted by the limited screen space. These examples also show that our system generally performs well for a wide range of K. The numbers of examples in the paper are reported in Table 2. The running time increases with the size of the dynamic set but for all the examples in the paper it takes less than 3 seconds and so is well suited for interactive exploration.

We demonstrate the effectiveness of our method by exploring various models of interest (Figs. 1, 11, 12). Liked (disliked) models specified by the user are highlighted in green (red). The user may also select some region of a model and express their preference. Models are placed in the 2D parametric space with important models (those the user likes most or most useful for active learning) shown bigger in the exploration. Only representative models are rendered and the user is also allowed to zoom in at some local region such that more models are revealed. For the initial exploration, the original orientation of the models is used for rendering as it is less reliable to establish correspondence between significantly different models. To provide more aesthetically pleasing rendering, the models in the repository have correct "up" direction, which may come from the original source, can be automatically computed [48] or specified manually when the repository is compiled. In later stages of interaction, the alignment between models are used so that they can rotate consistently. We use the selected model (for local active learning), or the model with the minimal overall diffusion distances to other models in τ (for global active learning) as reference and align all the other models to it.

Fig. 9 shows an example of exploring fighters. With a couple of active learning steps the rendered models are all fighters. However, alignment obtained using local correspondence works much better when the N-

Fig. 10. Results with different dynamic set sizes K. From left to right: results of global active learning with K being 50, 100, 300 and 400, respectively. Models are retrieved using the same sketch as Fig. 1. Top row: initial retrieval results with user preference; bottom row: results after active learning.

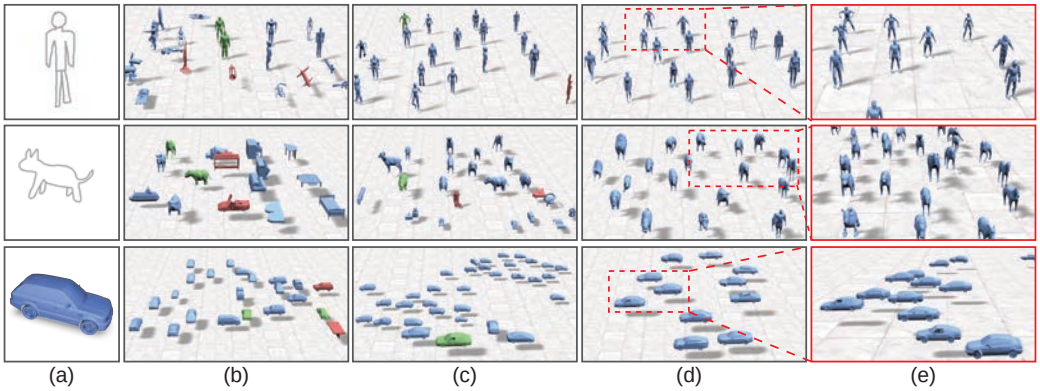


Fig. 11. Exploration of humans, quadrupeds and cars. (a) user input (drawn sketch/specified model),(b) initially retrieved models with liked (in green) and disliked (in red) models, (c)(d) results obtained with one and two iterations of global active learning, (e) zoom in.

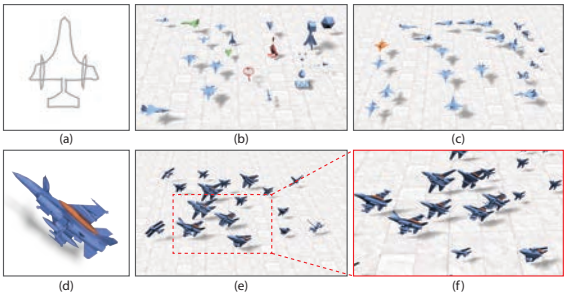


Fig. 12. Exploration of fighters. (a) user drawn sketch, (b) initially retrieved models with liked (in green) and disliked (in red) models, (c) result with global active learning, (d) selected local region highlighted (e) result of local active learning, (f) zoom in.

order graph is constructed with $N = 2$ (a) instead of $N = 5$ (b). Using $N = 2$ is also more robust than finding correspondence directly, as demonstrated by the example in (c) where direct alignment gives an inappropriate result (top arrow) whereas alignment via an intermediate model gives correct alignment (bottom arrow). Increasing N tends to reduce the number of added edges and $N = 2$ works well for all the examples in the paper.

Fig. 11 shows three active exploration examples, the first two (humans and quadrupeds) with sketch as input and the third (cars) using a model. The initial retrieval results contain irrelevant models, such as missiles for the human example, tables and beds (containing four legs) for the quadruped example. By using one or two iterations of active learning, relevant models are preserved and presented in a way with similar models being closer to each other, effectively giving an overview of relevant models. As local distances between neighboring models are used, our method is able to obtain models with moderately different poses or shapes (see e.g. humans). For the third example, although the input model (an SUV) is quite different from the models to be explored, the aim is achieved by using active learning with only a small number of labeled examples, demonstrating the effectiveness of active learning.

Figs. 1 and 12 show some examples of exploring chairs and fighters, with both global and local active learning. For the exploration of chairs, initial sketch based results involve irrelevant models such as a drum and a robot. With simple user interaction, non-chairs are removed. By using a local region, chair models with the similar handle shape are emphasized. For the fighter example, while the initially retrieved models are presented with relevant models clustered in close regions, they also contain irrelevant models such as a missile. Active learning is effective to select models of interest with a small amount of user interaction. The local region preference is more useful in this case to specify fighters with a similar secondary tank. The zooming in feature allows more relevant models to be displayed.

Retrieval comparison. Although the purpose of our work is exploration of large model repositories,

our method can be used for model retrieval and in this sense we quantitatively compare our work with other relevant feedback based retrieval methods [21], [22], [23], [24] on the Princeton Shape Benchmark which contains the ground truth labels. The work [25] uses Support Vector Machine (SVM) thus requires a training stage and the work [26] combines 10 different features many of which cannot be applied to models of poor connectivity. Thus these methods cannot be directly applied to the repositories considered in this work and are not suitable for direct comparison. Fig. 13 shows the average performance. Following the experimental set up in the previous work [24], we take the 7 categories with most models and use each model as input. Doing so avoids categories with too few models and thus user feedback would become too strong a constraint, oversimplifying the problem. For all the methods, correct manual labeling is provided as user preference for the top 20 returns. We then measure the Average Precision, First-Tier, Second-Tier and DCG (see [37]) (the larger the values, the better). Our active learning is more effective than these relevance feedback based approaches, because more detailed pairwise similarity is taken into account, as demonstrated by the results.

Comparison with [31]. Both [31] and our method use fuzzy correspondence for model exploration. [31] uses co-analysis to obtain robust local correspondence; their method however is designed to handle relatively small datasets as the complexity increases significantly with large datasets. Our method scales well and is able to handle repositories with more than $100K$ models in different categories.

We performed quantitative comparison with [31] using all the two published datasets with ground truth (Chair and Boeing) where Euclidean distances are used, as shown in Fig. 14. We take a random model from each dataset (as shown in the figure) and calculate the correspondence between this model and all the remaining models in the dataset. For key points, the deviations from the ground truth in the Euclidean metric are calculated (as used for evaluation in [31]). The graphs give the proportion (y-axis) of key points in percentage whose deviation is within a given threshold (x-axis). Although much simpler and more efficient, our method achieves similar performance as [31]. As the dynamic set cannot be predetermined, even if [31] is applied to a subset similar to the size of τ, several minutes would be needed, as reported in Table 4, which is not sufficiently efficient for interactive exploration (although their purpose is different and benefits from pairwise correspondence which is not needed for exploration). Using N-order graph (with $N = 2$ by default) for correspondence propagation gives better results (and slightly faster) than directly ($N = 1$), as shown in Fig. 14 and Table 4.

User study. It is generally difficult to quantitatively evaluate a system for interactive exploration. A user study is one way to get at least some indicators. Since a comprehensive user study is beyond the scope of this paper, we focus on a simple task that asks the user to find a specific target model in the database, starting from some initial queries (2D sketches or representative 3D models), as shown in Fig. 15. The 12 participants in our user study applied (1) a feature-based ranking (direct) approach, (2) relevance feedback (R-F) [24] and (3) our approach. A counterbalanced approach is used that assigns tasks and methods in a random order to each subject to avoid a learning bias. The distribution of interaction times is illustrated in Fig. 16 and the statistics are shown in Table 3. On average our approach took the shortest time across all the tasks. Using analysis of variance (ANOVA) the p-values in Table 3 show that our approach is superior to RF-based approach at $p = 0.001$ level for three tasks and at $p = 0.01$ level for the Chair task.

Task	(a) human	(b) chair	(c) helicopter	(d) car
t_d	95.17	91.44	128.77	81.54
t_{RF}	90.99	70.25	103.35	65.84
t_o	52.77	40.41	64.01	43.21
p	2.08×10^{-5}	0.0090	1.82×10^{-5}	1.25×10^{-7}

TABLE 3
Statistics of user study. t_d, t_{RF} and t_o are the average interaction time (in seconds) of direct, relevance feedback and our approaches. p is the p-value of statistical analysis between our and RF approaches.

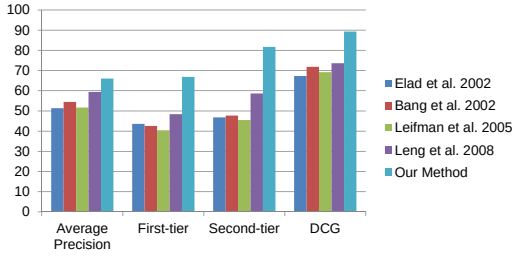


Fig. 13. Comparison of retrieval effectiveness with relevance feedback approaches (values in percentage).

Limitations. Our system has some limitations. The light field descriptors we used are insensitive to rigid transforms but may not be very effective for finding models under non-rigid deformation. Alternative signatures such as Heat Kernel Signatures [13] may work better in such cases but they may not be very robust for non-manifold models which is a common problem for man-made objects. As demonstrated in the first row of Fig. 11, our method is capable of finding smoothly deforming models because we only apply

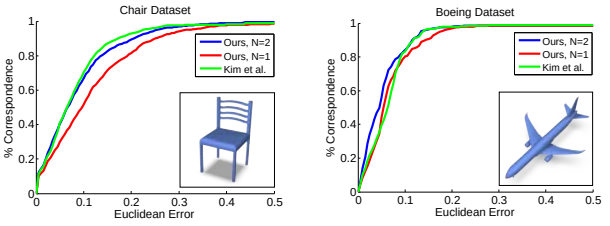


Fig. 14. Comparison of Euclidean error distribution with [31].

Dataset	Ours, N=1 (s)	Ours, N=2 (s)	Kim's[31] (s)
Chair	0.908	0.804	183.36
Boeing	0.804	0.679	155.40

TABLE 4
Running times of our and Kim's approaches.

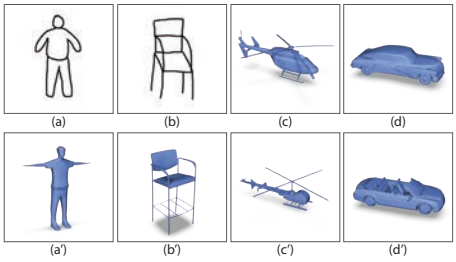


Fig. 15. Inputs/targets for user study. Top row: query sketches and models; bottom row: target models.

light field distances for sufficiently close models. As future work we would like to explore a combination of various signatures to handle diverse models more effectively. Feedback retrieval based on relative attributes similar to [49] could be used to allow more subtle personal preference to be expressed. New embedding techniques for visualization may also be explored [50].

7 Conclusion

In this paper, a novel active exploration algorithm is proposed for exploring large model repositories. As demonstrated by various examples, active learning is effective in obtaining relevant models with only a small amount of user input. Unlike traditional active learning, our approach maintains a dynamic set of models, allowing active learning to be efficient even on a very large set of models. We use both global and local geometric features to give the user flexibility of specifying their preference either on the whole models or in local regions of interest. We also propose novel visualization to clearly present the models in the dynamic set with the important models emphasized.

8 Acknowledgements

This work was supported by the National Basic Research Project of China (Project Number 2011CB302202), the Natural Science Foundation of China (Project Number 61120106007), the National High Technology Research & Development Program of China (Project Number 2012AA011801), Research Grant of Beijing Higher Institution Engineering Research Center, and Tsinghua University Initiative Scientific Research Program.

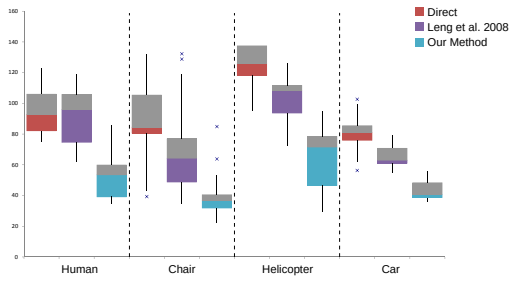


Fig. 16. Box-and-whisker plot of user interaction time (in seconds) to complete the browsing tasks.

References

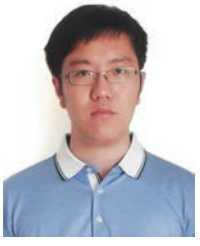
[1] "Google 3d warehouse," 3dwarehouse.sketchup.com.
[2] "Turbosquid," www.turbosquid.com.
[3] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools Appl.*, vol. 39, no. 3, pp. 441–471, 2008.
[4] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, 2002.
[5] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Symp. Geometry Processing*, 2003, pp. 156–164.
[6] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comp. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.
[7] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, and B. Guo, "Discriminative sketch-based 3d model retrieval via robust shape matching," *Comp. Graph. Forum*, pp. 2011–2020, 2011.
[8] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 31:1–31:10, 2012.
[9] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by example," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 652–663, 2004.
[10] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 130–150, 2006.
[11] Y. Liu, H. Zha, and H. Qin, "Shape topics: A compact representation and new algorithms for 3D partial shape retrieval," in *CVPR*, 2006, pp. 2025–2032.
[12] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape Google: Geometric words and expressions for invariant shape retrieval," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 1:1–1:20, 2011.
[13] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *SGP*, 2009, pp. 1383–1392.
[14] R. Wessel and R. Klein, "Learning the compositional structure of man-made objects for 3D shape retrieval," in *3DOR*, 2010, pp. 39–46.
[15] X. Xie, K. Xu, N. J. Mitra, D. Cohen-Or, and B. Chen, "Sketch-to-design: Context-based part assembly," *Comp. Graph. Forum*, vol. 32, no. 8, pp. 233–245, 2013.
[16] K. Xu, K. Chen, H. Fu, W.-L. Sun, and S.-M. Hu, "Sketch2scene: sketch-based co-retrieval and co-placement of 3d models," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 123:1–123:15, 2013.
[17] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML Workshop*, 2003, pp. 58–65.
[18] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *ACM Multimedia*, 2001, pp. 107–118.
[19] L. Wang, K. L. Chan, and Z. Zhang, "Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval," in *CVPR*, 2003, pp. 629–634.
[20] P. H. Gosselin and M. Cord, "Active learning methods for interactive image retrieval," *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1200–1211, 2008.
[21] M. Elad, A. Tal, and S. Ar, "Content based retrieval of VRML objects: an iterative and interactive approach," in *EG Workshop on Multimedia*, 2002, pp. 107–118.
[22] H. Bang and T. Chen, "Feature space warping: an approach to relevance feedback," in *ICIP*, 2002, pp. 22–25.
[23] G. Leifman, R. Meir, and A. Tal, "Semantic-oriented 3D shape retrieval using relevance feedback," *The Vis. Comp.*, vol. 21, no. 8-10, pp. 865–875, 2005.
[24] B. Leng and Z. Qin, "A powerful relevance feedback mechanism for content-based 3d model retrieval," *Multimedia Tools Appl.*, vol. 40, no. 1, pp. 135–150, 2008.

[25] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt, "Similarity learning for 3d object retrieval using relevance feedback and risk minimization," *IJCV*, vol. 89, no. 2-3, pp. 392–407, 2010.
[26] D. Giorgi, P. Frosini, M. Spagnuolo, and B. Falcidieno, "3D relevance feedback via multilevel relevance judgements," *The Vis. Comp.*, vol. 26, no. 10, pp. 1321–1338, 2010.
[27] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen, "Active co-analysis of a set of shapes," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 165:1–165:10, 2012.
[28] Q.-X. Huang, H. Su, and L. Guibas, "Fine-grained semi-supervised labeling of large shape collections," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 190:1–190:10, 2013.
[29] M. Ovsjanikov, W. Li, L. Guibas, and N. J. Mitra, "Exploration of continuous variability in collections of 3D shapes," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 33:1–33:10, 2011.
[30] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser, "Learning part-based templates from large collections of 3d shapes," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.
[31] V. G. Kim, W. Li, N. Mitra, S. DiVerdi, and T. Funkhouser, "Exploring collections of 3D models using fuzzy correspondences," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 54:1–54:11, 2012.
[32] Q. Huang, F. Wang, and L. Guibas, "Functional map networks for analyzing and exploring large shape collections," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 36:1–36:11, 2014.
[33] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 30:1–30:11, 2012.
[34] S.-S. Huang, A. Shamir, C.-H. Shen, H. Zhang, A. Sheffer, S.-M. Hu, and D. Cohen-Or, "Qualitative organization of collections of shapes via quartet analysis," *ACM Trans. Graph.*, vol. 32, no. 4, p. 71, 2013.
[35] Y. Kleiman, N. Fish, J. Lanir, and D. Cohen-Or, "Dynamic maps for exploring and browsing shapes," *Comput. Graph. Forum*, vol. 32, no. 5, pp. 187–196, 2013.
[36] J. O. Talton, D. Gibson, L. Yang, P. Hanrahan, and V. Koltun, "Exploratory modeling with collaborative design spaces," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 167:1–167:10, 2009.
[37] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *SMI*, 2004, pp. 167–178.
[38] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," in *Neural Information Processing Systems*, 2005, pp. 955–962.
[39] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Trans. Vis. Comp. Graph.*, vol. 17, no. 11, pp. 1624–1636, 2011.
[40] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *3DOR*, 2010, pp. 57–62.
[41] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.
[42] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. J. Guibas, "Functional maps: a flexible representation of maps between shapes." *ACM Trans. Graph.*, vol. 31, no. 4, p. 30, 2012.
[43] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[44] F. S. Hillier and L. G. J., *Introduction to Operations Research*. McGraw-Hill, 2001.
[45] "Tsinghua 3d model repositories," cg.cs.tsinghua.edu.cn/863.
[46] B. Li, T. Schreck, and A. G. *et al.*, "SHREC'12 track: Sketch-based 3D shape retrieval," in *3DOR*, 2012, pp. 109–118.
[47] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 73:1–73:12, 2009.
[48] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 42:1–42:7, 2008.
[49] A. Kovashka, D. Parikh, and K. Grauman, "WhittleSearch: Image Search with Relative Attribute Feedback," in *CVPR*, 2012, pp. 2973–2980.
[50] J. Wills, S. Agarwal, D. Kriegman, and S. Belongie, "Toward a perceptual space for gloss," *ACM Trans. Graph.*, vol. 28, no. 4, pp. 103:1–103:15, 2009.

Lin Gao received his BS degree in mathematics from Sichuan University and PhD degree in computer science from Tsinghua University. He is currently an assistant professor in Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics, geometric modeling and processing.

Yan-Pei Cao received his BS degree in computer science from Tsinghua University in 2013. He is currently a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics, geometric modeling and processing.

Yu-Kun Lai received his bachelors degree and PhD degree in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a lecturer of visual computing in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial board of The Visual Computer.



Hao-Zhi Huang received his BS degree in computer science from Tsinghua University in 2012. He is currently a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics, image processing and enhancement, image and video analysis.

Leif Kobbelt is a full professor and the head of the Computer Graphics Group at the RWTH Aachen University, Germany. His research interests include all areas of Computer Graphics and Geometry Processing with a focus on multi resolution and free-form modeling as well as the efficient handling of polygonal mesh data. He received his masters degree in (1992) and Ph.D. in (1994) from the University of Karlsruhe, Germany.

Shi-Min Hu received the PhD degree from Zhejiang University in 1996. He is currently a professor in the department of Computer Science and Technology, Tsinghua University, Beijing. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is associate Editor-in-Chief of The Visual Computer, associate Editor of Computer & Graphics and on the editorial board of Computer Aided Design. He is a member of the IEEE.